

# Centro de Investigación y Estudios Avanzados del I.P.N.



## Reconocimiento de 8 herramientas

Materia: Análisis de Imágenes Digitales

Catedrático: Dr. Wilfrido Gómez-Flores

Alumno: Luis Alberto Ballado Aradias

27 de agosto de 2023

# Índice

<b>1</b>	<b>Introducción</b>	<b>2</b>
<b>2</b>	<b>Adquisición de la imagen</b>	<b>3</b>
<b>3</b>	<b>Preprocesamiento</b>	<b>5</b>
<b>4</b>	<b>Segmentación</b>	<b>5</b>
<b>5</b>	<b>Extracción de características</b>	<b>6</b>
5.0.1	Geométricas básicas . . . . .	6
5.0.2	Rasgos basados en esqueleto . . . . .	6
5.0.3	Momentos invariantes Hue . . . . .	6
5.0.4	Cerco convexo . . . . .	7
<b>6</b>	<b>Clasificación</b>	<b>8</b>
<b>7</b>	<b>Resultados</b>	<b>9</b>
<b>8</b>	<b>Apendice</b>	<b>10</b>
8.1	Script de rasgos . . . . .	10
8.2	Script de data augmentation . . . . .	11
<b>9</b>	<b>Experimentando en python</b>	<b>12</b>
<b>10</b>	<b>Conclusiones</b>	<b>14</b>

## Lista de códigos

1	Script de rasgos . . . . .	10
2	Python example . . . . .	11
3	experimentación python . . . . .	12
4	Python example . . . . .	12

# 1. Introducción

El Análisis de Imágenes Digitales es un área fascinante que engloba muchas áreas de las ciencias computacionales como Optimización y Aprendizaje de máquinas.

El procesamiento de imágenes es un área multidisciplinaria que tiene como objetivo la manipulación y análisis de la información de una imagen en su parte más reducida conocida como pixel (picture element). Emergiendo una gran cantidad de técnicas y metodologías para el procesamiento y manipulación de imágenes.

Con el aumento de imágenes digitales, el procesamiento se ha vuelto necesario para aplicaciones en robótica, que la segmentación de imágenes puede facilitar la navegación para un robot [4], en el diagnóstico de cáncer ayuda a los médicos en dar diagnósticos más acertados [3]. Asimismo, en la parte del entretenimiento con implementación de realidad aumentada [1] .

La detección y reconocimiento que son parte del área de visión por computadora, se han vuelto cada vez más utilizadas en aplicaciones como traductores de idiomas, análisis médicos, reconocimiento de rostros faciales.

La detección y categorización de objetos con imágenes se descompone de los siguientes procesos:

1. **Adquisición de la imagen.-** La toma de fotografía se realiza mediante una cámara que cuente con un sensor que puede ser de tipo CMOS, el sensor convierte la captura de la escena en señales digitales que son guardadas como una imagen.
2. **Preprocesamiento.-** Mediante las técnicas existentes se busca el mejoramiento del contraste, realzar características reduciendo el ruido, para hacer de la imagen lo más limpia posible para el paso de segmentación.
3. **Segmentación.-** Es la obtención de las regiones de interés que ayuda a la extracción de características.
4. **Extracción de características.-** Este proceso busca describir en cierta forma la imagen, describiendo valores de forma geométrica, colores, texturas y valores invariantes que se mantienen a pesar de rotaciones o escalamientos.
5. **Clasificación.-** Asigna una clase a cada región de interés dentro de la imagen basados en los rasgos extraídos en el paso anterior.

El proyecto busca el reconocimiento de ocho herramientas, aplicando las técnicas vistas en clase con el uso de MATLAB.

La figura 1 muestran las etapas a desarrollar para el reconocimiento de las ocho herramientas.

Una imagen  $f(x, y)$  se puede modelar como el producto de dos componentes  $f(x, y) = i(x, y) \cdot r(x, y)$ , donde  $0 < i(x, y) < \infty$  es la componente de iluminación y  $0 < r(x, y) < 1$  es la componente de reflectancia.

Las dificultades que se tienen dentro del procesamiento de imágenes digitales son la iluminación del ambiente, siendo los sensores perceptibles a la iluminancia al obtener la imagen.

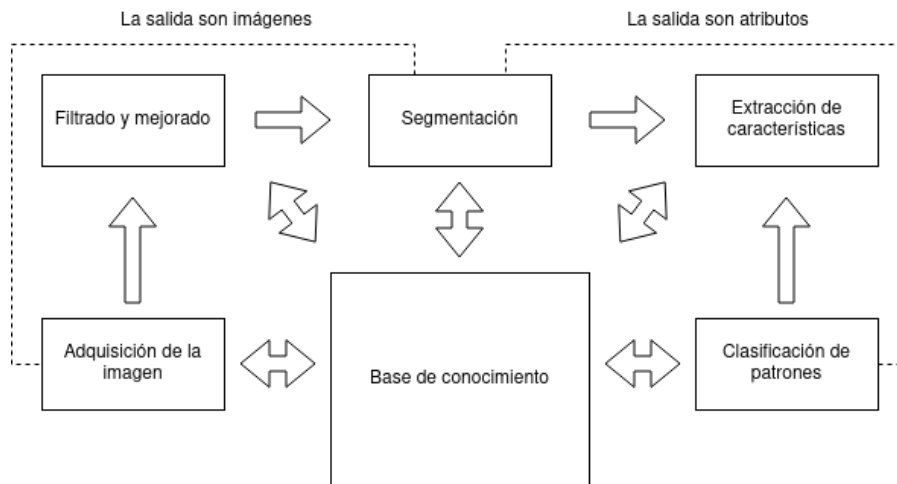


Figura 1: Etapas básicas para el análisis de imágenes

## 2. Adquisición de la imagen

La adquisición de las imágenes se realizó con las siguientes herramientas:

- Cámara web Logitech C920s HD Pro 1080p webcam
- Cubo para la toma de fotografías
- Control de iluminación



Figura 2: Dimmer para el ajuste de la iluminación

La cámara web cuenta con un sensor de tipo CMOS de 2-mega pixel [2], obteniendo imágenes

limpias y muy bien contrastadas.

El objetivo del proyecto, es el reconocer automáticamente ocho herramientas que son:

- Martillo
- Desarmador
- Cinta de medir
- Llave perica
- Tijeras
- Pinza de punta
- Pinza eléctricas
- Pinza de presión

La toma de fotografías se realiza en un ambiente controlado como se muestra en la figura 3, ajustando la iluminación con un dimmer mostrado en la figura 2.



Figura 3: Cubo para la adquisición de imágenes

Se usaron transformaciones geométricas para modificar la relación espacial entre píxeles en una imagen.

Usando propiedades que no deformen al objeto de interes como:

- Rotaciones
- Traslaciones

Logrando crear imágenes artificiales, de modo tal que sirvan para el aumento del conjunto de imágenes.

### 3. Preprocesamiento

Al contar con imágenes de buena calidad, sólo se mejoro el contraste para reducir las sombras que las herramientas pueden formar y lograr obtener el contorno que representa más la imagen a reconocer.

El mejoramiento como manipulación de una imagen tal que el resultado sea más útil que la imagen original para una aplicación particular.

Una transformación de intensidad modifica el constraste.

Ecualización del histograma.

No se tuvo que ecualizar ni mejorar por CLAHE, pero es bueno saber la existencia de estas tecnicas para crear sistemas y filtros sofisticados.

Modo de degradación de una imagen.

No se hizo uso de ningun filtro, buscando evitar el efecto de deslavado y poder detectar los bordes lo más puros posibles.

Pero el filtro de difusión anisotrópico puede considerarse en la implementación.

### 4. Segmentación

Se hacen pruebas con diversas técnicas de segmentación, optando por un resultado sencillo en base a operaciones morfologicas en base de una imagen de detección de bordes a partir de canny edges.

Contorno de una región

El contorno de una región conexa es el conjunto de pixeles que tienen al menos en pixel vecino que corresponde al fondo en 4 ó 8 adyacencia.

Buscar que tipo de etiquetado usa BWLABEL .- QUE ALGORITMO?

Etiquetado de regiones conexas

## 5. Extracción de características

El reconocimiento automático de objetos requiere calcular rasgos que describan propiedades físicas de los objetos.

Un rasgo (atributo o característica), es un valor numérico que cuantifica alguna propiedad de forma, textura, color, geometría, etc.

- Un rasgo debe ser discriminante, invariante, incorrelado y rápido de computar.
- Los rasgos de forma generalmente se calculan a partir de la segmentación del objeto y se divide entre todos los píxeles de las regiones de interés.
- Basados en región distribución de todos los píxeles.
- Basados en contorno, variaciones a lo largo del contorno.

La extracción de características que se utilizaron en el proyecto son las siguientes:

1. Rasgos geométricos básicos
2. Momentos de Hue
3. Cerco convexo
4. Esqueletización

### 5.0.1. Geométricas básicas

- Redondez
- Circularidad
- Compacidad
- Factor de forma

El área es el número total de píxeles que cubre la región del objeto.

El perímetro es la longitud del contorno del objeto. El resultado depende del tipo de conectividad que pueda tener.

### 5.0.2. Rasgos basados en esqueleto

- Puntos finales
- Ramas
- Número de píxeles

### 5.0.3. Momentos invariantes Hue

Los momentos son proyecciones de una función sobre una base polinomial usados para medir la distribución de masa de un cuerpo.

#### 5.0.4. Cerco convexo

Los momentos son proyecciones de una función sobre una base polinomial usados para medir la distribución de masa de un cuerpo.

- Solidez
- Convexidad



## 6. Clasificación

Etapas del reconocimiento de objetos en imágenes.

1. Sensado.- Capturar la imagen.
2. Preprocesamiento.- Mejorar la calidad de la imagen y segmentar los objetos de interes.
3. Extracción de características.- Describir los objetos con rasgos cuantitativos discriminantes e invariantes.
4. Clasificación.- Asignar una etiqueta de clase a cada objeto.

Obtener un vector de patrones.

Para el proyecto se consideran ocho clases sendo representadas por los objetos siguientes:

1. Cinta de medir
2. Desarmador
3. Martillo
4. Pinza electrica
5. Pinza de presión
6. Pinza de punta
7. Perica
8. Tijeras

En este proyecto se usa el método de datos de entrenamiento que son vectores de patrones asociados a una etiqueta de clase  $(x,y)$  con  $y \in \Omega = w_1, ..., w_c$

### MOSTRAR EN DIAGRAMA DE LOS PROCEOS PARA LA CLASIFICACION

La selección del algoritmo de clasificación debe tener las siguientes cualidades.

- Generación de fronteras de decisión no lineales.
- Clasificación en más de dos clases (multiclasas).
- Entrenamiento en un tiempo de cómputo razonable.

Se elige el clasificador k-nn de vecinos más cercanos por su simpleza.

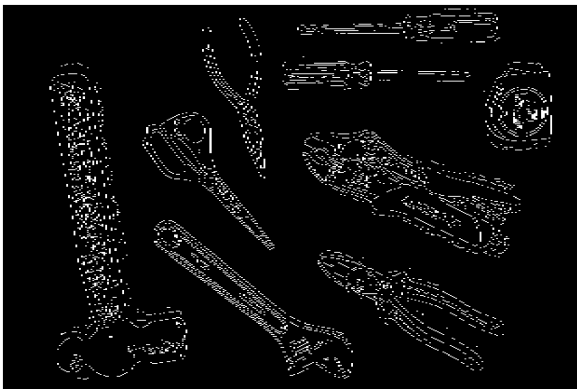
## 7. Resultados



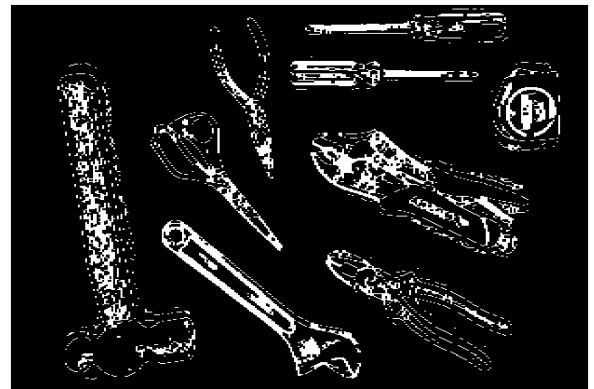
(a)



(b)



(c)



(d)

Figura 4: My composed figure

Dentro de las imagenes, la similitud de esqueletos entre la cinta, la llave perica, tijera, pinzas de punta y electrica que tienen formas similares.

La mayoría de desarmadores no tiene ramas y tiene dos puntos finales, lo que lo confunde con una cinta al solo considerar las ramas y los puntos terminales se tiene una exactitud del 20 %, mientras al agregar el num de pixeles crece al 78 %, agregando más información como el área y perimetro la probabilidad de acierto crece aún más, presentandose valores arriba del 90 %.

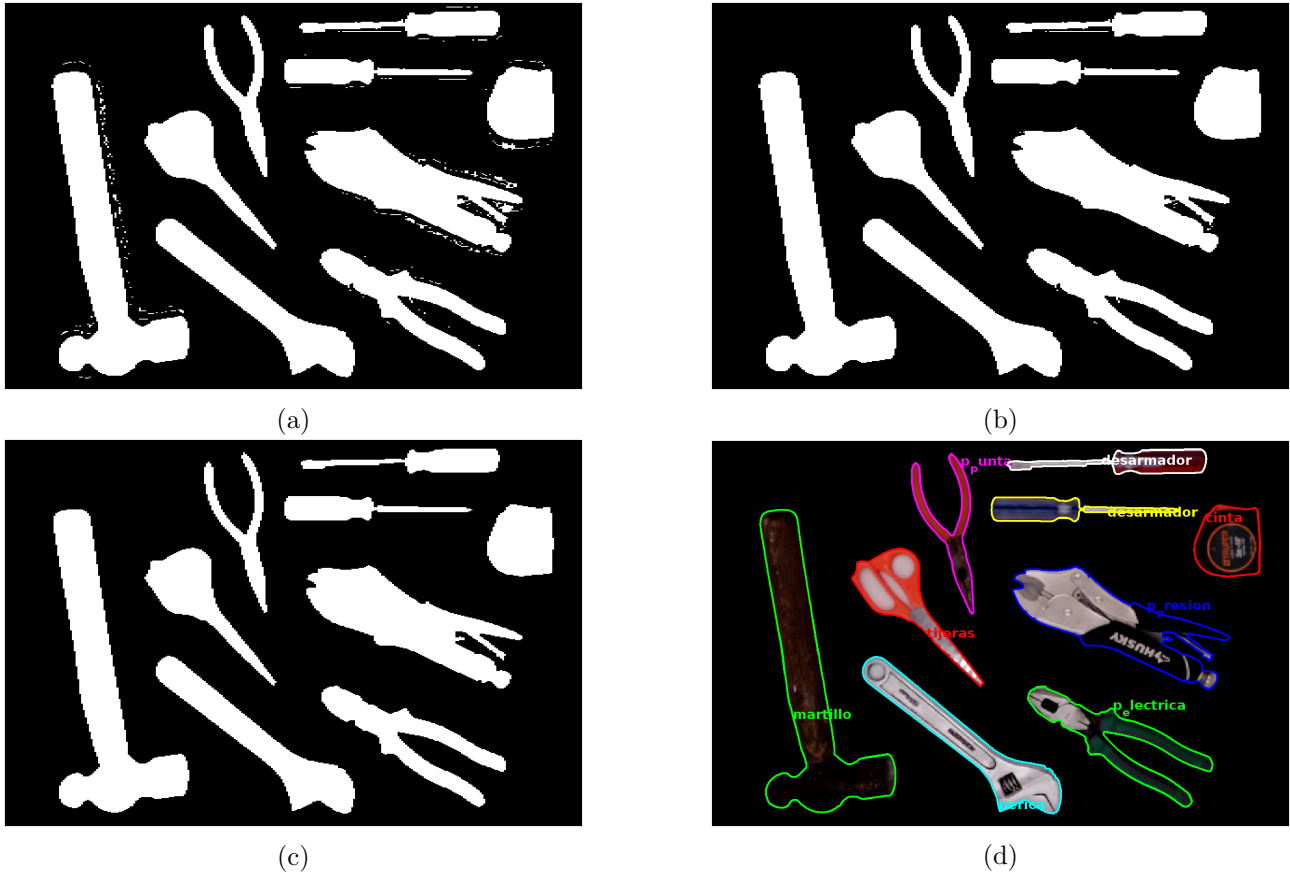


Figura 5: My composed figure

## 8. Apendice

### 8.1. Script de rasgos

La obtención de rasgos se realizó mediante un script que itera entre todas las carpetas de las imágenes, haciendo rotaciones y traslaciones aleatorias, buscando mantener la imagen, por lo que se le añade un efecto para que el fondo se replique

```

1 %% Sample Matlab code
2 !mv test.txt test2.txt
3 A = [1, 2, 3;... foo
4       4, 5, 6];
5 s = 'abcd';
6 for k = 1:4
7     Disp(s(k)) % bar
8 end
9 %{
10 create row vector x, then reverse it
11 %}
12 x = linspace(0,1,101);
13 y = x(end:-1:1);

```

Listing 1: Script de rasgos

## 8.2. Script de data aummentation

We did some experiments ...

```
1 %% Sample Matlab code
2 !mv test.txt test2.txt
3 A = [1, 2, 3;... foo
4      4, 5, 6];
5 s = 'abcd';
6 for k = 1:4
7     Disp(s(k)) % bar
8 end
9 %{
10 create row vector x, then reverse it
11 %}
12 x = linspace(0,1,101);
13 y = x(end:-1:1);
```

Listing 2: Python example

## 9. Experimentando en python

```
1 import numpy as np
2
3 def incmatrix(genl1,genl2):
4     m = len(genl1)
5     n = len(genl2)
6     M = None #to become the incidence matrix
7     VT = np.zeros((n*m,1), int) #dummy variable
8
9     #compute the bitwise xor matrix
10    M1 = bitxormatrix(genl1)
11    M2 = np.triu(bitxormatrix(genl2),1)
12
13    for i in range(m-1):
14        for j in range(i+1, m):
15            [r,c] = np.where(M2 == M1[i,j])
16            for k in range(len(r)):
17                VT[(i)*n + r[k]] = 1;
18                VT[(i)*n + c[k]] = 1;
19                VT[(j)*n + r[k]] = 1;
20                VT[(j)*n + c[k]] = 1;
21
22            if M is None:
23                M = np.copy(VT)
24            else:
25                M = np.concatenate((M, VT), 1)
26
27            VT = np.zeros((n*m,1), int)
28
29    return M
```

Listing 3: experimentación python

```
1 import numpy as np
2
3 def incmatrix(genl1,genl2):
4     m = len(genl1)
5     n = len(genl2)
6     M = None #to become the incidence matrix
7     VT = np.zeros((n*m,1), int) #dummy variable
8
9     #compute the bitwise xor matrix
10    M1 = bitxormatrix(genl1)
11    M2 = np.triu(bitxormatrix(genl2),1)
12
13    for i in range(m-1):
14        for j in range(i+1, m):
15            [r,c] = np.where(M2 == M1[i,j])
16            for k in range(len(r)):
17                VT[(i)*n + r[k]] = 1;
18                VT[(i)*n + c[k]] = 1;
19                VT[(j)*n + r[k]] = 1;
20                VT[(j)*n + c[k]] = 1;
21
22            if M is None:
23                M = np.copy(VT)
24            else:
25                M = np.concatenate((M, VT), 1)
26
27            VT = np.zeros((n*m,1), int)
```

```
28  
29     return M
```

Listing 4: Python example

## 10. Conclusiones

Estudios tempranos de Inteligencia Artificial buscaban duplicar los pensamientos humanos, pero ahora los estudios recientes muestran la tendencia en replicar el resultado y las computadoras actúan como sistemas expertos.

Las computadoras son dispositivos simbólicos capaces de manipular cualquier tipo de símbolos siendo los números una clase importante, pero las computadoras son más generales que eso. Sabemos de la generalidad de la computación desde los tiempos de Alan Turing en los 1930's y se tienen intuiciones que Babbage tuvo también estudios que fueron afirmados por Ada Lovelace, en 1842 Ada Lovelace escribió sobre la ingeniería analítica de Babbage que buscaba unir el mundo mecánico con el mundo de las cosas abstractas, en la psicología moderna es llamado The physical symbol system hypothesis y son las bases en que la Inteligencia Artificial funciona.

La Inteligencia Artificial como ciencia emplea el uso de computadoras para procesar conocimientos simbólicos usando métodos de inferencias lógicas, en otras palabras, nos referimos a inferencia y no cálculos que se piensa en el sentido tradicional, hablamos de conocimiento y no números como en la forma tradicional.

Al poner conocimiento en programas computacionales que para los humanos nos es fácil o a veces un reto intelectual y el conocimiento que le pasamos es representativo en cierta forma particular.

Por otra parte, los sistemas expertos basados en conocimientos son aplicables en cualquier área en que sea especializado el conocimiento y sea rutinario la toma de decisiones, estrategias para resolver problemas, diagnósticos, .. etc.

Siendo de gran ayuda en un gran rango de aplicaciones, el **conocimiento médico**, en clasificación de un experto, puede estar reflejado con el funcionamiento de una red neuronal.

Para algún diagnóstico médico o segmentadores más fáciles, pero igual de complicados como la segmentación en imágenes para la creación de sistemas autónomos.

Programas que se puedan ejecutar en cualquier computadora o dispositivo que permita la ejecución de código multiplataforma como python o C++, que tienen capacidades altas de manipulación simbólica, la toma de decisiones es primordial en una inteligencia basada en conocimiento y esa manipulación simbólica es necesaria.

## Referencias

- [1] R. Aggarwal and A. Singhal. Augmented reality and its effect on our life. In *2019 9th International Conference on Cloud Computing, Data Science and Engineering (Confluence)*, pages 510–515, 2019.
- [2] CDW. logitech c920s HD. <https://www.cdw.com/product/logitech-hd-pro-webcam-c920s-webcam/5479466>, 2023. [Online; accessed 27-Agust-2023].
- [3] M. N. R. Devi, A. Kumar, G. Swetha, U. S. Chavan, and V. M. Davasam. Cancer detection using image processing and machine learning. In *2022 International Conference on Artificial Intelligence and Data Engineering (AIDE)*, pages 96–100, 2022.
- [4] N. Smolyanskiy, A. Kamenev, J. Smith, and S. Birchfield. Toward low-flying autonomous mav trail navigation using deep neural networks for environmental awareness, 2017.